Decodificación

La decodificación consiste en habilitar un dispositivo, normalmente una memoria o registros de un periférico, dentro de rango de direcciones específicas del mapa de memoria de un procesador. El circuito que realiza esta tarea se le conoce como decodificador.

Para entender como trabaja el circuito decodificador definiremos el concepto de mapa de memoria.

Mapa de memoria
Es todo el espacio de direcciones que puede generar o manejar un procesador con su bus de direcciones. Generalmente el tamaño del Contador de Programa determina el bus de direcciones.

Para realizar el diseño de un circuito decodificador seguiremos cuatro sencillos pasos:

Paso 1. Determinar el tamaño del bus de direcciones y datos de la(s) memoria(s) a partir de su organización. También, con el bus de direcciones del procesador determinar el tamaño del mapa de memoria.

Paso 2. Dibujar el procesador y la(s) memoria(s) con sus buses de datos y direcciones y realizar la conexión de los bits de los buses uno a uno. Después de hacer la conexión, determinar cuales son los bits “sobrantes” del bus de direcciones del procesador.

Paso 3. Determinar la dirección inicial y final del rango dentro del mapa de memoria del procesador donde se desea mapear o habilitar la(s) memoria(s).

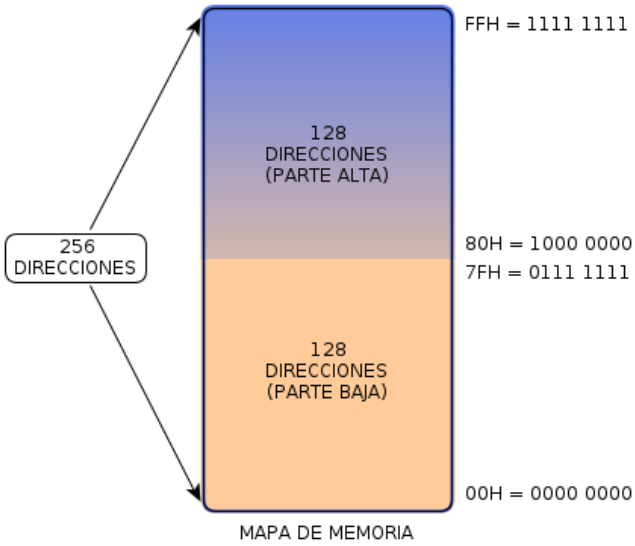
Paso 4. Ver que valor toman los bits sobrantes del bus de direcciones del procesador en el rango donde se desea mapear. Colocar la compuerta (AND ó NOR) decodificadora para que la memoria sea habilitada en el rango deseado.

Ejemplo 1:

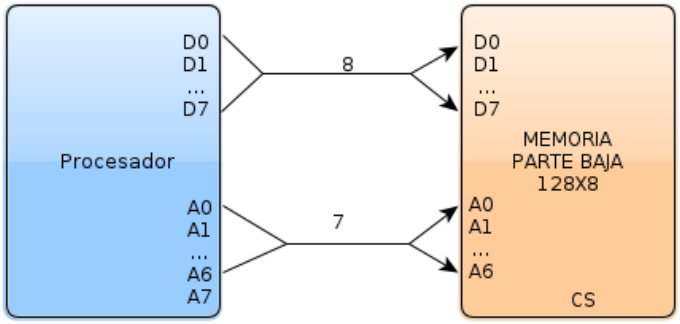
Considere un procesador con 8 bits en el bus de direcciones y 8 bits en el bus de datos. Usando memorias con una organización de 128x8, diseñe un circuito decodificador que permita habilitarla en la parte baja del mapa de memoria.

Paso 1. La memoria tiene una organización de 128x8, el primer número (128) me dice cuantas localidades o direcciones tiene la memoria y el segundo número (8) el tamaño del dato que puede guardar la memoria en cada dirección. Para este caso se tienen 128 direcciones de memoria, por lo que $2^7=128$. Esto quiere decir que se tienen 7 bits en el bus de direcciones de la memoria. Por otro lado el bus de datos de la memoria es de 8 bits.

Como el procesador tiene un bus de direcciones de 8 bits, puede generar $2^8 = 256$ direcciones de memoria, por lo que el mapa de memoria es el siguiente:



Paso 2. Si queremos conectar una memoria con una organización de 128x8 en la parte baja del mapa de memoria, debemos realizar primero la conexión del bus de datos del procesador con el bus de datos de la memoria y el bus de direcciones del procesador con el bus de direcciones de la memoria, de la siguiente forma:



Si observamos el bit A7 del procesador queda desconectado, es con ese bit con el que vamos a diseñar el circuito decodificador.

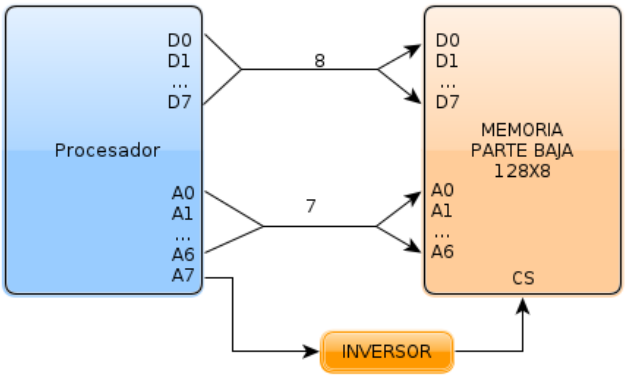
Paso 3. Debemos determinar la dirección inicial y final del rango donde queremos mapear la memoria En este caso se quiere mapear en la parte baja del mapa de memoria, por lo que la dirección inicial del rango comienza en la dirección 00H y la dirección final la vamos a obtener sumando el tamaño de la memoria, 128. Como $2^7 = 128 = 0111\ 1111$ (7 unos) = 7FH. Por lo que la dirección final es:

$$\begin{array}{r} 00H \\ + \\ 7FH \\ \hline 7FH \end{array}$$

Paso 4. Para determinar el valor que toma el bit A7 del procesador en el rango deseado (la parte baja del mapa de memoria), colocamos en binario la dirección inicial y final del rango:

	Rango	A7	A6	A5	A4	A3	A2	A1	A0
Dirección inicial	00H	0	0	0	0	0	0	0	0
Dirección final	7FH	0	1	1	1	1	1	1	1

Se puede observar que en la parte baja del mapa de memoria el bit A7 tiene el valor 0. Si queremos que la memoria se habilite en el rango de direcciones de la parte baja del mapa de memoria, debemos conectar un inversor entre el bit A7 del procesador y CS de la memoria. Tal como se muestra a continuación:



La ecuación para la señal CS de la memoria es CS = (NOT A7).

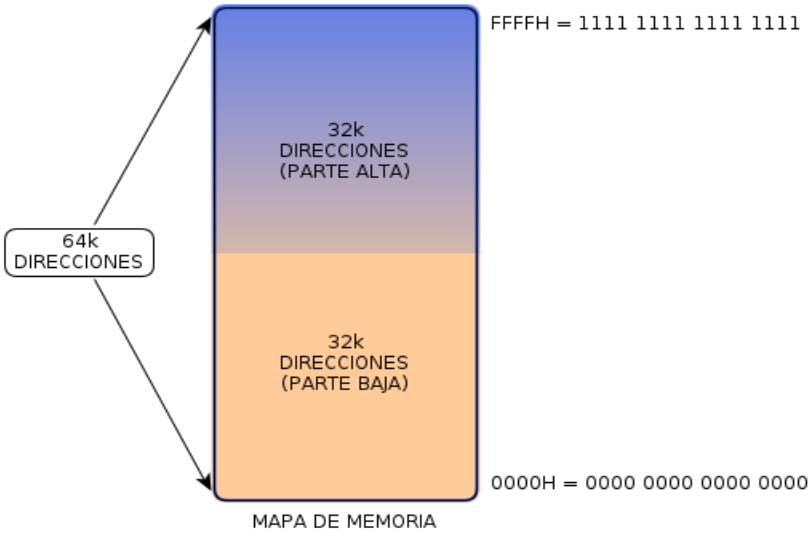
Ejemplo 2:

Considere un procesador con 16 bits en el bus de direcciones y 16 bits en el bus de datos.

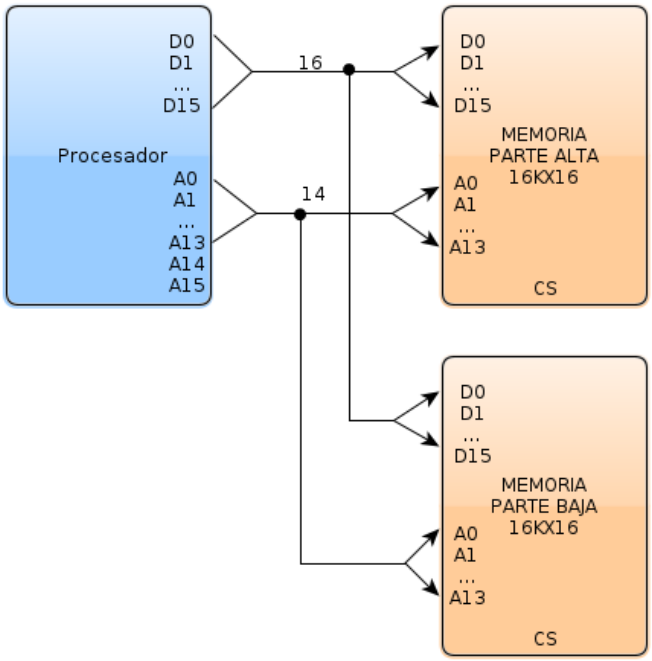
a) Usando dos memorias con una organización de 16Kx16, diseñe un circuito decodificador que permita habilitar una de ellas en la parte baja del mapa de memoria y otra en la parte alta.

Paso 1. La memoria tiene una organización de 16Kx16, el primer número (16K) me dice cuantas localidades o direcciones tiene la memoria y el segundo número (16) el tamaño del dato que puede guardar la memoria en cada dirección. Para este caso se tienen 16K direcciones de memoria, por lo que 2^{14}=16384=16K. Esto quiere decir que se tienen 14 bits en el bus de direcciones de la memoria. Por otro lado el bus de datos de la memoria es de 16 bits.

Como el procesador tiene un bus de direcciones de 16 bits, puede generar 2^{16} = 65536 = 64K direcciones de memoria, por lo que el mapa de memoria es el siguiente:



Paso 2. Si queremos conectar memorias con una organización de 16kx16 en la parte baja y alta del mapa de memoria, debemos realizar primero la conexión del bus de datos del procesador con el bus de datos de las memorias y el bus de direcciones del procesador con el bus de direcciones de las memorias, de la siguiente forma:



Si observamos el bit A14 y A15 del procesador quedan desconectados, es con esos bits con los que vamos a diseñar el circuito decodificador.

Paso 3. Debemos determinar la dirección inicial y final de los rangos donde queremos mapear las memorias. En este caso el mapa de memoria que tenemos en este procesador es de 64K, por lo que podemos dividirlo en 4 rangos de 16K. *Aunque solo nos interesan dos rangos, vamos a analizar todos.*

La dirección inicial del rango 1 comienza en la dirección 0000H y la dirección final la vamos a obtener sumando el tamaño de la memoria, 16K. Como 2^{14} = 16K = 0011 1111 1111 1111 (14 unos) = 3FFFH. Por lo que la dirección final es:

```
  0000H
+
  3FFFH
 ------
  3FFFH
```

La dirección inicial del rango 2 comienza en la dirección 4000H=3FFFH+1 y la dirección final la vamos a obtener sumando el tamaño de la memoria, 16K. Como 2^{14} = 16K = 0011 1111 1111 1111 (14 unos) = 3FFFH. Por lo que la dirección final es:

```
  4000H
+
  3FFFH
 ------
  7FFFH
```

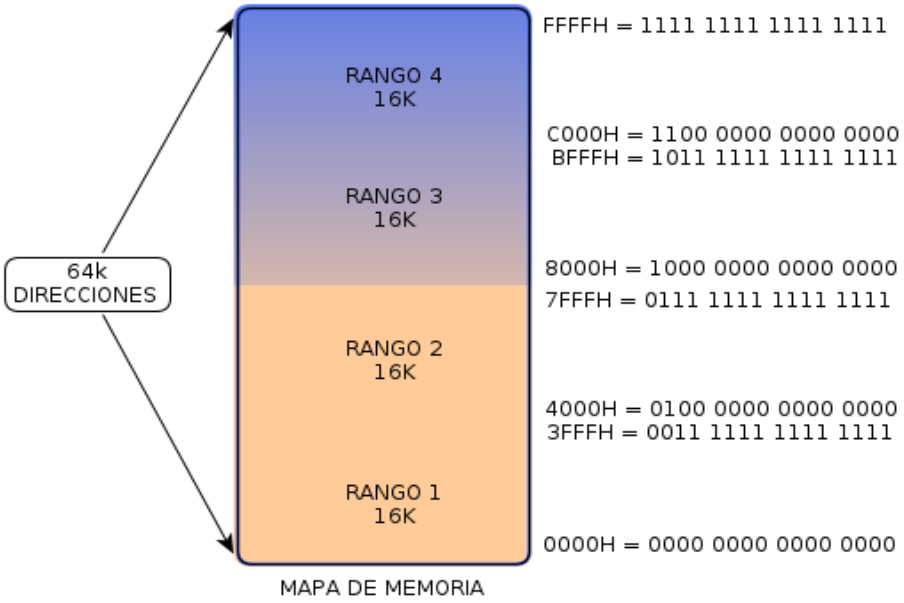
La dirección inicial del rango 3 comienza en la dirección 8000H=7FFFH+1 y la dirección final la vamos a obtener sumando el tamaño de la memoria, 16K. Como 2^{14} = 16K = 0011 1111 1111 1111 (14 unos) = 3FFFH. Por lo que la dirección final es:

```
  8000H
+
  3FFFH
 ------
  BFFFH
```

La dirección inicial del rango 4 comienza en la dirección C000H=BFFFH+1 y la dirección final la vamos a obtener sumando el tamaño de la memoria, 16K. Como 2^{14} = 16K = 0011 1111 1111 1111 (14 unos) = 3FFFH. Por lo que la dirección final es:

```
  C000H
+
  3FFFH
 ------
  FFFFH
```

El mapa de memoria con los rangos de 16K se observa en la siguiente figura:



Paso 4. Las memorias deben mapearse en el Rango 1 y Rango 4, por lo que debemos analizar los bits A15 y A14 en esos rangos, (*aquí analizamos todos los rangos*):

Rango 1	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Dirección inicial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0000H	0				0				0				0			
Dirección final	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3FFFH	3				F				F				F			

Rango 2	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Dirección inicial	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4000H	4				0				0				0			
Dirección final	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7FFFH	7				F				F				F			

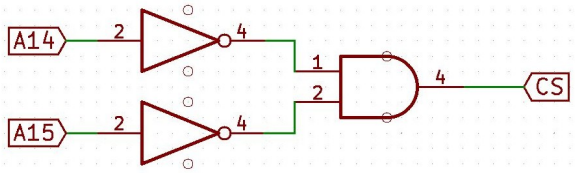
Rango 3	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Dirección inicial	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8000H	8				0				0				0			
Dirección final	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
BFFFH	B				F				F				F			

Rango 4	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Dirección inicial	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C000H	C				0				0				0			

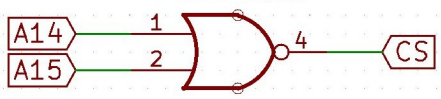
Autor: Victor Hugo García Ortega

Dirección final	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
FFFFH	F				F				F				F			

Como podemos observar en el rango 1 los bits A15 = 0 y A14 = 0, por lo que el circuito decodificador lo podemos diseñar con una compuerta AND1 o con una compuerta NOR1 de la siguiente manera:

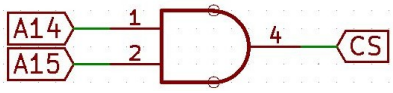


La ecuación para la señal CS de la memoria usando la compuerta AND es:
CS = (NOT A15) AND (NOT A14).

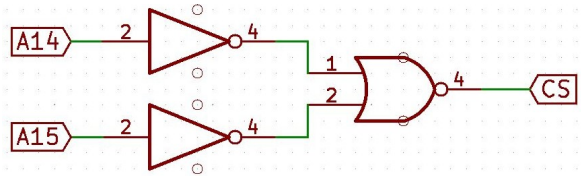


La ecuación para la señal CS de la memoria usando la compuerta NOR es:
CS = (A15) NOR (A14).

Para el rango 4 los bits A15 = 1 y A14 = 1, por lo que el circuito decodificador lo podemos diseñar con una compuerta AND4 o con una compuerta NOR4 de la siguiente manera:

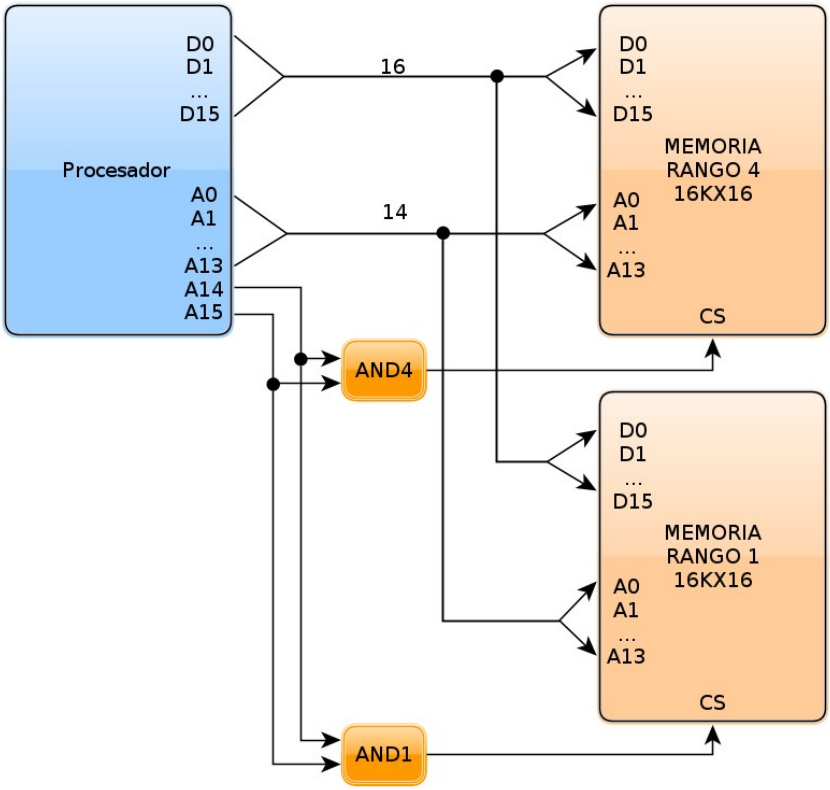


La ecuación para la señal CS de la memoria usando la compuerta AND es:
CS = (A15) AND (A14).



La ecuación para la señal CS de la memoria usando la compuerta NOR es:
CS = (NOT A15) NOR (NOT A14).

Finalmente el circuito completo para la decodificación de memorias se muestra en la figura siguiente:



Tarea:

Considere un procesador con 16 bits en el bus de direcciones y 16 bits en el bus de datos.

Usando una memoria con una organización de 4Kx16, diseñe un circuito decodificador que permita habilitar la memoria en la dirección 5000H.

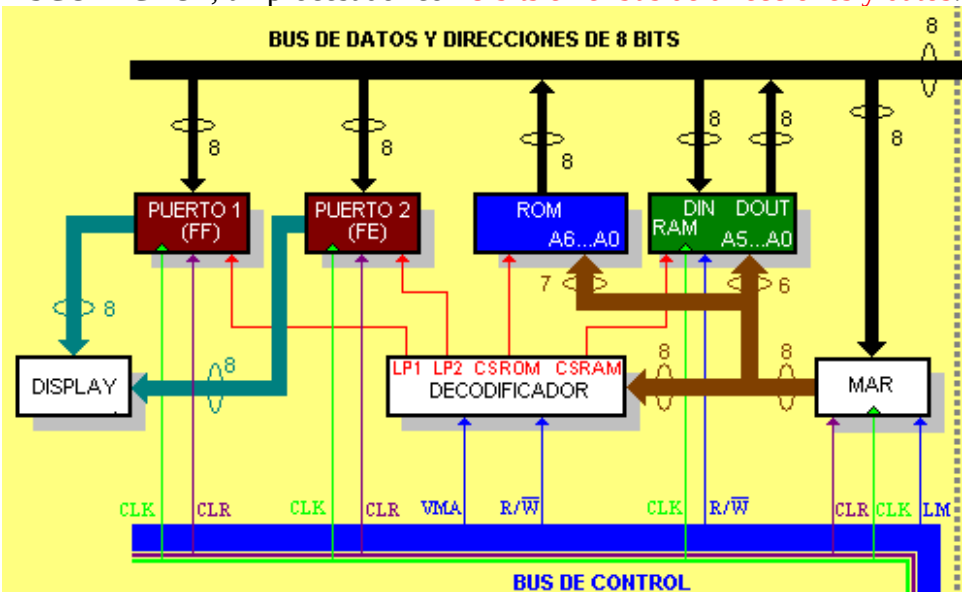
Tarea:

Considere un procesador con 16 bits en el bus de direcciones y 16 bits en el bus de datos.

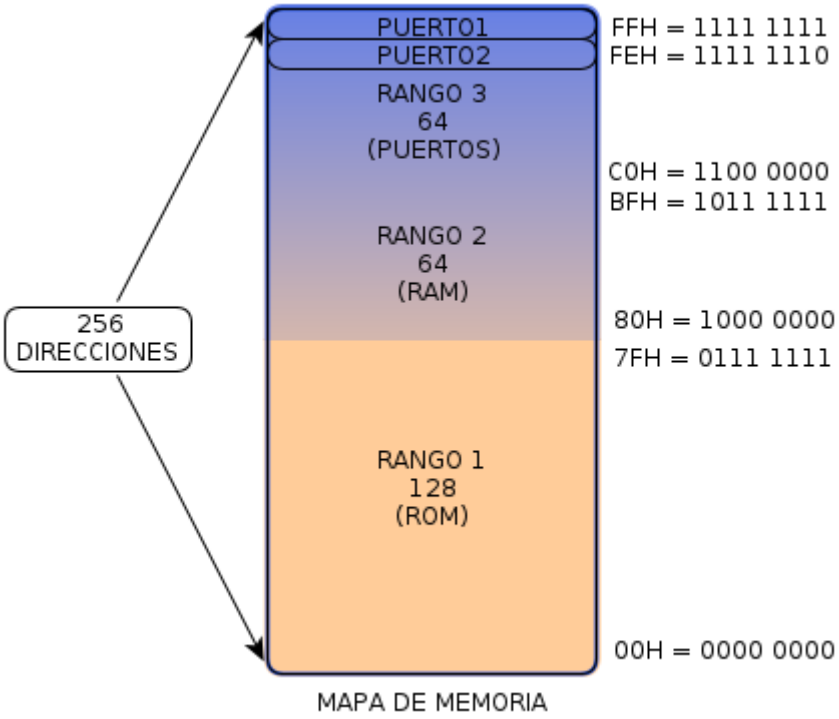
b) Usando una memoria con una organización de 2Kx16, diseñe un circuito decodificador que permita habilitar la memoria en la dirección A800H.

Ejemplo 3:

Considere el ESCOMICRO1, un procesador con 8 bits en el bus de direcciones y datos.



El mapa de memoria mapea una memoria ROM, una memoria RAM y dos puertos de salida.



a) Usando una memoria ROM con una organización de 128x8, diseñe un circuito decodificador que permita habilitarla en la parte baja del mapa de memoria.

Este ejemplo muestra el mismo procesador del ejemplo 1, en este ejemplo la ecuación debe considerar las señales de control VMA (Valid Memory Address) y RW (Read/Write).

Autor: Victor Hugo García Ortega

La señal VMA le dice al decodificador que trabaje (decodifique)
VMA = 1, el decodificador trabaja
VMA = 0, el decodificador no trabaja

La señal RW me dice si vamos a realizar una escritura o lectura
RW = 1, operación de lectura
RW = 0, operación de escritura

La ecuación para la memoria ROM es:

CSROM = (NOT A7) AND (RW) AND (VMA)

b) Usando una memoria RAM con una organización de 64x8, diseñe un circuito decodificador que permita habilitarla a partir de la dirección 80H.
2^6 = 64, 0011 1111 = 3FH

80H
+
3FH

BFH

	Rango	A7	A6	A5	A4	A3	A2	A1	A0
Dirección inicial	80H	1	0	0	0	0	0	0	0
Dirección final	BFH	1	0	1	1	1	1	1	1

CSRAM = (A7) AND (NOT A6) AND (VMA)

c) Usando un registro (puerto 1), diseñe un circuito decodificador que permita habilitarlo en la dirección FFH = 1111 1111.

LP1 = (A7) AND (A6) AND (A5) AND (A4) AND (A3) AND (A2) AND (A1) AND (A0)
AND (NOT RW) AND (VMA)

d) Usando un registro (puerto 2), diseñe un circuito decodificador que permita habilitarlo en la dirección FEH = 1111 1110.

LP2 = (A7) AND (A6) AND (A5) AND (A4) AND (A3) AND (A2) AND (A1) AND (NOT A0)
AND (NOT RW) AND (VMA)

Paso 1. La memoria tiene una organización de 128x8, el primer número (128) me dice cuantas localidades o direcciones tiene la memoria y el segundo número (8) el tamaño del dato que puede guardar la memoria en cada dirección. Para este caso se tienen 128 direcciones de memoria, por lo

que 2^7=128. Esto quiere decir que se tienen 7 bits en el bus de direcciones de la memoria. Por otro lado el bus de datos de la memoria es de 8 bits.

Como el procesador tiene un bus de direcciones de 8 bits, puede generar 2^8 = 256 direcciones de memoria, por lo que el mapa de memoria es el siguiente: